

# Routing electric vehicles with a single recharge per route

Maximilian Löffler<sup>1</sup> | Guy Desaulniers<sup>2</sup> | Stefan Irnich<sup>3</sup>  | Michael Schneider<sup>1</sup>

<sup>1</sup>Deutsche Post Chair—Optimization of Distribution Networks, RWTH Aachen University, Aachen, Germany

<sup>2</sup>Department of Mathematics and Industrial Engineering, GERAD and Polytechnique Montréal, Montreal, Canada

<sup>3</sup>Chair of Logistics Management, Gutenberg School of Management and Economics, Johannes Gutenberg University Mainz, Mainz, Germany

## Correspondence

Stefan Irnich, Chair of Logistics Management, Gutenberg School of Management and Economics, Johannes Gutenberg University Mainz, Mainz, Germany.

Email: irnich@uni-mainz.de

## Funding information

This research was supported by the Deutsche Forschungsgemeinschaft (DFG), Grant/Award Numbers: IR122/7-1, SCHN 1497/1-1

## Abstract

Driven by environmental considerations, regulations on vehicle emissions, and the offer of major subsidies, electric commercial vehicles (ECVs) are receiving ever stronger attention in logistics companies. Route planning for ECV fleets requires consideration of the special characteristics of ECVs, like limited driving range and the potential need to recharge en route at dedicated recharging stations. From a practical viewpoint, the number of recharge operations of each vehicle can very often be restricted to one recharge per route because (i) typical route distances in the most important application areas of ECVs, like small package shipping and food or beverage distribution, do not require more than one recharge given the current driving range of ECVs, and (ii) operations managers are very reluctant to plan vehicle routes with two or more recharges because recharging operations are perceived as unproductive idle times. We develop a simple hybrid of large neighborhood search and granular tabu search to solve the resulting electric vehicle-routing problem with time windows and single recharge (EVRPTWS), considering the possibility of both full and partial recharge. The heuristic works on routes represented as customer sequences, and recharge operations are implicitly considered by determining the recharging position in the route, the recharging station to visit, and the amount to be recharged in optimal fashion. We discuss how our algorithm can be extended to handle nonlinear recharging times, different recharging times per station, and time-dependent waiting times at stations. In numerical studies on EVRPTWS instances from the literature, the method provides optimal or near-optimal solutions for instances with up to 100 customers within reasonable runtimes. Additional studies investigate the cost savings potential of partial recharges in comparison to full recharges in the presence of time-window constraints, and examine the factors that influence this cost saving potential.

## KEYWORDS

electric vehicles, granular tabu search, implicit representation, large neighborhood search, time windows, vehicle routing

## 1 | INTRODUCTION

*Electric commercial vehicles* (ECVs) produce only minimal noise and have no local greenhouse gas emissions. Rising energy costs, new regulations concerning the emission of greenhouse gases, and a rising environmental awareness render ECVs an alternative to conventional internal combustion vehicles, although they are currently not competitive with regards to costs. These conditions have triggered green initiatives at companies of the transportation sector, for example, in the field of small-package

This is an open access article under the terms of the Creative Commons Attribution License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

© 2020 The Authors. *Networks* published by Wiley Periodicals LLC.

shipping [20] and the distribution of food [24]. Moreover, governments and private companies have started to provide the required infrastructure to pave the way for this electrification trend [17,36].

The planning of the last-mile delivery operations of ECVs requires models and algorithms for solving *vehicle-routing problems* (VRPs, [38]) that incorporate the special characteristics of ECVs: a limited driving range and the potential need to recharge en route at dedicated recharging stations. Such VRPs have recently been proposed, covering different subsets of the practically relevant VRP constraints (vehicle capacity, time windows, maximal route duration, heterogeneous vehicle fleet) and different assumptions on energy consumption, recharging infrastructure, and the recharging procedure (for an overview, see [7,19,25,31]). In particular, several real-world features have been studied, among them hybrid propulsion technologies [16], detailed models of charging and discharging processes and their impact on battery lifetime degradation [26,27], uncertainty in energy consumption [28], and capacitated recharging stations [12].

Almost all papers in the literature assume that multiple recharges are possible on each vehicle route (exceptions are the works of Desaulniers et al. [5,6]). However, practical considerations can well motivate to restrict recharge operations to a single recharge per route: major application areas of ECVs lie in the fields of small package shipping and goods distribution in a city logistics context. In these cases, distances of routes are typically rather short, for example, Kleindorfer et al. [20] report that the lengths of the routes of the French national post operator, La Poste, are in the range of 100 to 150 km. Estimates for ECV driving ranges vary significantly in the literature within a range of roughly 100 to 200 km [9,25,39] and external factors like cold temperature, range anxiety, battery age, and grades reduce the originally reported ranges. Nevertheless, from these figures, it can be assumed that a very high percentage of the routes in the typical ECV application areas can be covered with not more than one recharge (see also [8]). New developments in battery technology and steadily decreasing battery prices substantiate this assumption. Moreover, even if fast-charging modes are used, real-life recharging times are relatively long with 30 minutes and above [25]. This renders tours with more than one recharge unattractive in many practical situations because recharging operations are perceived as unproductive idle times. So, although multiple recharges per route are very interesting from a computational viewpoint, they are less important from a practical one.

Motivated by these observations, the paper at hand studies the *electric VRP with time windows and single recharge* (EVRPTWS), originally introduced by Desaulniers et al. [5]. The EVRPTWS is a variant of the basic EVRPTW proposed by Schneider et al. [32] in which at most one recharge per vehicle route is allowed. Depending on the type of recharge, we distinguish the EVRPTWS with *single partial* (SP) recharge and with *single full* (SF) recharge. Using the former policy (SP), the amount of energy to recharge is a decision variable and it is possible to stop the recharging process before the maximal capacity of the battery is reached, while in the latter policy (SF), the battery is always reloaded to its maximal capacity. The EVRPTWS also differs from the original EVRPTW with regard to the objective function. EVRPTWS considers the classical VRP objective of minimizing traveled distance, contrary to the hierarchical objective used in EVRPTW: minimize the number of employed vehicles first, minimize traveled distance second. We made this choice for two reasons. First, to maintain comparability between exact approaches (usually using the classical objective) and our metaheuristic approach. Second, to be better able to quantify the benefits of allowing partial recharges: the two goals of minimizing the number of employed vehicles and the traveled distance are sometimes conflicting. This may lead to distorted results when comparing the solutions with partial and full recharges. Because the EVRPTWS extends the VRPTW, this problem is NP-hard. Thus, the available exact methods consume considerable runtimes and are still not able to consistently solve large-scale instances within acceptable runtimes (see [6]).

The contributions of our paper are:

- We present a hybrid of *large neighborhood search* (LNS) and *granular tabu search* (GTS), called LNS  $\times$  GTS, which is, to the best of our knowledge, the first metaheuristic for the EVRPTWS. From a methodological viewpoint, we contribute in the direction of developing simpler heuristics with convincing solution quality, something that the community is advocating in recent years: The design of the LNS  $\times$  GTS algorithm is kept simple, and several known algorithmic components are combined in a new and successful fashion. Contrary to most existing approaches for routing ECVs, our algorithm uses an implicit representation of recharging visits: recharging visits are not explicitly contained in the encoding of a solution, and no operators are used to change the position of a recharging visit within or between routes. Instead, the operators of our algorithm directly work on sequences of customers, and the position of the recharging visit in a route and the amount of energy to recharge (in case of partial recharges) are determined in optimal fashion. The computational studies show that our method provides optimal or near-optimal solutions for instances with up to 100 customers within reasonable runtimes.
- Although several real-life features have recently been studied in EVRP research (see above), we consider a basic EVRP variant to be able to assess the quality of our algorithm by means of a comparison using the results of the exact approaches of Desaulniers et al. [5,6]. However, we provide a detailed explanation of how our approach can be extended to handle the following important real-world extensions: nonlinear charging times [11,22], chargers with different recharging rates [8], and time-dependent waiting times at recharging stations [19].

- Additional studies concentrate on the cost savings of partial recharges in comparison to full recharges in the presence of time windows. We investigate which problem characteristics (size, distribution of customer locations, time window density, and time window width) influence these cost benefits. Note that Desaulniers et al. [5] have studied the effect of partial recharges for the EVRPTWS; however, (i) the results are based only on the subset of the benchmark instances that could be solved to optimality and may therefore be biased towards a certain problem structure, and (ii) no analysis of the factors influencing the benefits of partial recharges is performed. Other works in the literature (e.g., [18]) only address the multiple-recharge version of EVRPTW and use a hierarchical objective that first minimizes the number of employed vehicles.

The remainder of this paper is organized as follows: Section 2 provides a detailed problem description. The tailored LNS  $\times$  GTS approach for the EVRPTWS is described in Section 3. The numerical studies to investigate the questions raised above are detailed in Section 4. We close the paper with final conclusions in Section 5.

## 2 | THE ELECTRIC VEHICLE-ROUTING PROBLEM WITH TIME WINDOWS AND SINGLE RECHARGE

We define the two variants SP (partial recharges allowed) and SF (full recharges required) of the EVRPTWS. An instance of the EVRPTWS can be defined with the help of a directed graph  $G = (V, A)$ , where the vertex set  $V$  comprises the *customers*  $N = \{1, \dots, n\}$ , the *recharging stations*  $R$ , and the *depot* 0. For each customer  $i \in N$ , let  $d_i$  be the demand,  $[e_i, \ell_i]$  the time window describing the possible service start time, and  $s_i$  the service duration. The planning horizon is given by the depot time window  $[e_0, \ell_0]$ . For the depot and all recharging stations, we assume service times and demands of zero, that is,  $s_i = d_i = 0$  for  $i \in \{0\} \cup R$ . Moreover, the recharging stations  $i \in R$  have a nonconstraining time window  $[e_i, \ell_i]$ .

The arcs  $(i, j) \in A$  represent the possible travel connections with a travel time  $t_{ij}$  and a routing cost  $c_{ij}$ . The battery-capacity constraint is modeled with the help of the time  $b_{ij}$  required to recharge the consumed energy when traveling between locations  $i$  and  $j$  for each arc  $(i, j) \in A$  (see [5]). Let  $B$  be the battery capacity, expressed in time units needed to fully recharge an empty battery. We assume that the triangle inequality holds true for the costs  $c_{ij}$  and travel times  $t_{ij}$  as well as recharging times  $b_{ij}$ .

A fleet of homogeneous vehicles each with capacity  $Q$  is hosted at the depot 0. A feasible route in the general EVRPTW is a cycle through 0 satisfying the recharging, vehicle-capacity, time-window, and battery-capacity constraints. To make feasibility conditions explicit, let the route be  $\omega = (i_0, i_1, \dots, i_p)$  for some  $p > 1$  and  $i_0 = i_p = 0$ . For the single-recharge policy, at most one of the vertices  $i_0, i_1, \dots, i_p$  can be a recharging station, say  $i_q = r \in R$ , while  $i_1, \dots, i_{q-1}, i_{q+1}, \dots, i_{p-1} \in N$ , that is, the vertices are customers. Obviously, the route fulfills the capacity constraint if  $\sum_{j=0}^p d_{i_j} \leq Q$ . For the two other constraints, one must decide on the schedule and the amount to recharge at every visited recharging station. In particular, the corresponding recharging times need to be incorporated into the time-window constraints. A necessary condition for a route to be feasible is that there exists a schedule  $(\tau_0, \tau_1, \dots, \tau_p)$  and a (battery-)recharging plan  $(y_0, y_1, \dots, y_p)$  that fulfill the following conditions: First, recharging is only possible at a recharging station, that is,  $y_j = 0$  if  $i_j \in N \cup \{0\}$  and  $0 \leq y_j \leq B$  for  $i_j \in R$ . Second, the time-window constraints are fulfilled, that is,  $\tau_j \in [e_{i_j}, \ell_{i_j}]$  for all  $j = 0, \dots, p$  as well as  $\tau_{j-1} + s_{i_{j-1}} + y_{j-1} + t_{i_{j-1}, i_j} \leq \tau_j$  for all  $j = 1, \dots, p$  assuming that there are no service times at recharging stations, that is,  $s_{i_j} = 0$  for  $i_j \in R$ . Third, the recharging plan must be feasible, that is,  $B - \sum_{j=1}^k b_{i_{j-1}, i_j} + \sum_{j=1}^{k-1} y_j \geq 0$  and  $B - \sum_{j=1}^k b_{i_{j-1}, i_j} + \sum_{j=1}^k y_j \leq B$  for all  $k = 1, \dots, p$ .

For the EVRPTWS with SP policy, the given conditions are sufficient. For the EVRPTWS with SF policy, the battery must always be completely recharged at stations, that is,  $B - \sum_{j=1}^q b_{i_{j-1}, i_j} + \sum_{j=1}^q y_j = B$  if  $i_q \in R$ .

Note that we have intentionally given a slightly more abstract but also universal definition than needed for the EVRPTWS. Indeed, definitions also hold true for the multiple-recharge policies (MP, MF), in which several visits to recharging stations are allowed, that is,  $i_1, \dots, i_{p-1} \in R \cup N$  (see [5]). In all cases, a feasible solution to the EVRPTW (with any policy SP, SF, MP, MF) is a set of feasible routes that together visit and serve all customers  $i \in N$  exactly once. A feasible solution is optimal if it minimizes the total cost of the routes, where the cost of a route  $\omega = (i_0, i_1, \dots, i_p)$  is  $\sum_{j=1}^p c_{i_{j-1}, i_j}$ .

For the EVRPTWS variants, the feasibility check can be simplified as follows: for routes that do not contain a recharging station, time-window and capacity constraints are like in the VRPTW, while the battery-capacity check is that  $\sum_{j=1}^p b_{i_{j-1}, i_j} \leq B$  holds. For routes that do contain a single recharging station  $r \in R$  at position  $q$ , that is, a route  $(i_0, \dots, i_q = r, \dots, i_p)$  with  $0 < q < p$ , the battery-capacity check is that  $\sum_{j=1}^q b_{i_{j-1}, i_j} \leq B$  and  $\sum_{j=q+1}^p b_{i_{j-1}, i_j} \leq B$  holds. For the time-window feasibility, we can first determine the recharging time  $y_q$  at station  $i_q = r$ , which is exactly

$$y_q = \max \left\{ 0, B - \sum_{j=1}^p b_{i_{j-1}, i_j} \right\} \quad \text{and} \quad y_q = \sum_{j=1}^q b_{i_{j-1}, i_j} \quad (1)$$

for SP and SF, respectively. Then, any (standard) time-window check known for the VRPTW can be applied using new service times  $s_i + y_j$  for all  $j = 0, \dots, p$  (defining  $y_j = 0$  for all  $j \neq q$ ).

### 3 | LNS × GTS FOR SOLVING EVRPTWS

In a very first step, our method preprocesses EVRPTWS instances by removing all infeasible arcs, that is, arcs that cannot be contained in a solution that is feasible with respect to load capacity and time windows, using rules analogous to the ones presented in Schneider et al. [32].

Algorithm 1 shows a pseudocode of the LNS × GTS algorithm: Initially, we use an insertion heuristic to generate an initial solution  $\Sigma$ , which is feasible with respect to time windows, vehicle capacity, and battery capacity. The heuristic iteratively either inserts a customer into an existing route by evaluating all insertion points for every remaining customer or creates a new route with a single customer, depending on which alternative leads to the smaller increase in cost. In the improvement phase, LNS and GTS alternate. For this phase, we use the implicit representation of recharging visits that we detail in Section 3.1. Solutions in the improvement phase may be infeasible with respect to time-window and vehicle-capacity constraints but have to be feasible regarding battery capacity. Infeasible solutions are handled by a dynamic penalty mechanism which uses the generalized cost function  $f(\Sigma)$  explained in Section 3.2. The main purpose of the LNS phase is diversification (see Section 3.3), while the GTS phase is used for intensification of the search (see Section 3.4).

---

#### Algorithm 1: LNS × GTS

---

```

1  $\Sigma \leftarrow \text{insertionHeuristic}()$ 
2  $\Sigma' \leftarrow \Sigma$ 
3 repeat
4   /* LNS phase */
5    $\delta \leftarrow \text{drawNumberOfCustomersToRemove}([0.2n, 0.8n])$ 
6    $\Sigma' \leftarrow \text{randomDestroy}(\Sigma', \delta)$ 
7    $\Sigma' \leftarrow \text{Regret/GRASP-Repair}(\Sigma')$ 
8   /* GTS phase */
9    $\Sigma' \leftarrow \text{granularTabuSearch}(\Sigma')$ 
10  /* acceptance criterion */
11  if  $f(\Sigma') < f(\Sigma)$  then
12     $\Sigma \leftarrow \Sigma'$ 
13 until  $\Sigma$  not improved for  $\eta_{lms}$  iterations

```

---

#### 3.1 | Implicit recharging decision

Our algorithm uses an implicit representation of the recharge decisions, that is, the objects manipulated by our search operators are *vertex sequences* (cycles through 0) in the induced graph  $G[N \cup \{0\}]$ . From now on, we explicitly distinguish between vertex sequences  $\sigma$  and their associated routes  $\omega = \omega(\sigma)$ , which result from inserting (if necessary) a recharging station visit in an optimal manner. For every vertex sequence, we can easily determine whether a recharge is necessary. If so, optimal decisions need to be made regarding (i) the recharging position in the sequence, (ii) the recharging station to use, and (iii) the amount of energy to recharge. These optimal decisions have to be carried out for feasibility checks and for computing the costs of a vertex sequence  $\sigma$ .

The idea of an implicit representation used together with a decoder is not new (see, e.g., [29]) and has been applied before in the context of VRPs with intermediate stops (see, e.g., [14,21]). A dynamic programming component for determining the recharging decision for a given customer sequence in optimal fashion has been introduced by Hiermann et al. [15] for a variant of the EVRPTW and has been used by many algorithms for related problems since then.

To efficiently compute the optimal recharging decisions, we can restrict the set of potential recharging stations that we have to consider for recharging between the tail and head vertex of a given arc. For a given vertex sequence, we can also restrict the positions in the sequence where we have to test for a feasible recharge option. These steps are detailed in the following.

##### 3.1.1 | Restrict relevant recharging stations for a given arc

For each arc  $(v, w) \in A$  with  $v, w \notin R$  that is not deemed infeasible by the preprocessing, we can determine the set of potential recharging stations at the position between vertices  $v$  and  $w$ .

First, a visit of the station has to be possible without causing a constraint violation. A recharging station  $r \in R$  is *feasible* for  $(v, w)$  if the route  $\omega = (0, v, r, w, 0)$  does not violate the battery-capacity, time-window, or vehicle-capacity constraints. Second, a recharging station only has to be considered for a visit between  $v$  and  $w$  if it is not dominated by any other recharging station.

Formally, a recharging station  $r_1 \in R$  *dominates* a station  $r_2 \in R$  if all of the following conditions hold:

$$c_{vr_1} + c_{r_1w} \leq c_{vr_2} + c_{r_2w}, \quad (2a)$$

$$b_{vr_1} \leq b_{vr_2}, \quad (2b)$$

$$b_{r_1w} \leq b_{r_2w}, \quad (2c)$$

$$\max\{e_w, e_v + s_v + t_{vr_1} + y_{r_1} + t_{r_1w}\} \leq \max\{e_w, e_v + s_v + t_{vr_2} + y_{r_2} + t_{r_2w}\}, \quad (2d)$$

$$\min\{\ell_v, \ell_w - t_{r_1w} - y_{r_1} - t_{vr_1} - s_v\} \leq \min\{\ell_v, \ell_w - t_{r_2w} - y_{r_2} - t_{vr_2} - s_v\}, \quad (2e)$$

where (2a) compares the cost of detours, (2b) the battery consumption to station, (2c) the battery consumption from station, (2d) the earliest start time of service at  $w$ , and (2e) the latest start time of service at  $v$ . In the two latter formulas, the values  $y_{r_1}$  and  $y_{r_2}$  are computed via (1) for the routes  $(0, v, r_1, w, 0)$  and  $(0, v, r_2, w, 0)$ , respectively.

In the following, the set of all feasible, nondominated *potential recharging stations* for the arc  $(v, w) \in A$  is denoted by  $R(v, w)$ .

### 3.1.2 | Restrict the recharge positions for a given vertex sequence

For a vertex sequence  $\sigma = (i_0, i_1, \dots, i_p)$  with  $i_0 = i_p = 0$  and  $i_1, i_2, \dots, i_{p-1} \in N$ , we can skip the determination of the best recharge decision if either

$$\sum_{j=1}^p b_{i_{j-1}i_j} > 2B \quad \text{or} \quad \sum_{j=1}^p b_{i_{j-1}i_j} \leq B \quad (3)$$

holds: in the first case the route is certainly infeasible due to a total energy consumption that is too large, and in the second case, no recharging is required.

Otherwise, we can restrict the potential positions  $j$  in the sequence at which a feasible recharge between  $i_{j-1}$  and  $i_j$  can take place. For this purpose, we define

$$j_{\min} = \arg \min_{j \in \{1, \dots, p\}} \left\{ \sum_{k=j+1}^p b_{i_{k-1}i_k} \leq B \right\} \quad \text{and} \quad j_{\max} = \arg \max_{j \in \{1, \dots, p\}} \left\{ \sum_{k=1}^{j-1} b_{i_{k-1}i_k} \leq B \right\} \quad (4)$$

as the minimal and maximal positions, respectively. Note that even if  $\sum_{j=1}^p b_{i_{j-1}i_j} \leq 2B$  (neither of the conditions in (3) may be fulfilled), the detour caused by a potential recharging station can lead to a route that is battery-capacity infeasible at the end. A more precise check is therefore needed, as explained in the following.

### 3.1.3 | Generalized cost function

The purpose of the *generalized cost function* is 2-fold: first, it exactly distinguishes between battery-feasible and infeasible routes. Recall that battery-infeasible routes are not permitted in LNS  $\times$  GTS (see Algorithm 1). However, solutions in the improvement phase may be infeasible with respect to time-window and vehicle-capacity constraints. Therefore, the second purpose of the generalized cost function is the computation of a penalty value for solutions that do not adhere to the capacity or time-window constraints.

Regarding the vehicle-capacity constraints, a route  $\omega = (i_0, i_1, \dots, i_p)$  serving customers  $i_1, \dots, i_{p-1}$  overloads the vehicle by  $\max\{0, Q - \sum_{j=1}^{p-1} d_{i_j}\}$  units. Accordingly, we penalize solutions with  $\alpha \cdot \max\{0, Q - \sum_{j=1}^{p-1} d_{i_j}\}$ , where  $\alpha$  is a dynamic penalty factor, explained in Section 3.2.

Regarding the time-window constraints, the penalty mechanism is much more sophisticated. We again distinguish between routes (containing a specific recharging station at a fixed position) and vertex sequences.

At the route level, that is, when the recharging station, its position, and the amount to recharge are known, we use a time-warp like penalty function (see, e.g., [40]) summarized in Algorithm 2. More precisely, we assume for a route  $\omega = (i_0, i_1, \dots, i_p)$  that service times at customers and the recharging time at the recharging station (if any) are given by the values  $(S_0, S_1, \dots, S_p)$ . An as-early-as-possible schedule is constructed (propagating schedule times in forward direction), where in case of a time-window violation the actual violation is recorded, and the schedule time is reduced to the time-window upper bound (last feasible point in time). The final penalty is the sum of the violations weighted with a second dynamic penalty factor  $\beta$ .

**Algorithm 2:** Penalty  $P_{tw}(\omega, (S_j))$ 


---

**input :** route  $\omega = (i_0, i_1, \dots, i_p)$  and service plus recharging times  $(S_0, S_1, \dots, S_p)$   
**output:** time window violation  $\geq 0$

- 1  $P_{tw} \leftarrow 0$  // penalty value
- 2  $\tau \leftarrow e_{i_0}$  // schedule time
- 3 **for**  $j = 1, 2, \dots, p$  **do**
- 4      $\tau \leftarrow \max\{e_{i_j}, \tau + S_{i_{j-1}} + t_{i_{j-1}, i_j}\}$
- 5     **if**  $\tau > \ell_{i_j}$  **then**
- 6          $P_{tw} \leftarrow P_{tw} + (\tau - \ell_{i_j})$
- 7          $\tau \leftarrow \ell_{i_j}$
- 8 **return**  $P_{tw}$

---

At the vertex-sequence level, the idea is to very quickly determine all undominated battery-feasible associated routes. This is established by exploiting the knowledge about the relevant undominated recharging stations per arc and restricting the recharging positions as explained in the two previous subsections. For each constructed route, the recharging time (amount to recharge) is computed first, before the penalty function for the route is evaluated. The pseudocode of Algorithm 3 provides an overview of the generalized cost function  $f_{gen}$ . The Steps 1 to 21 are specific for the EVRPTWS, while Steps 23-24 cover the standard VRPTW case. The outer nested loop (Steps 9-18) tests the restricted recharge positions. Once this position is known, the accumulated recharging times  $b^1$  and  $b^2$  on the segments before and after the recharging arc can be computed (Step 8). These values are updated in constant time via Step 18. The inner loop (Steps 10-17) tests specific admissible recharging stations for the respective arc. The Steps 19 and 20 cover the cases of battery-infeasible vertex sequences for which the quick check (3) is not precise enough.

It should be noted that the implicit recharging decision depends on the value of  $\beta$ . For a sufficiently small  $\beta$ , even a recharging decision which leads to a violation of the time-window constraints might be preferred over a recharging decision which completely maintains feasibility with respect to time windows at the expense of a longer detour. Figure 1 demonstrates this effect with an example for the case of partial recharges. The arc labels denote the routing cost, and we assume  $c_{ij} = t_{ij} = b_{ij}$ . The time window of customer  $i$  is given in square brackets as  $[e_i, \ell_i]$ , and all customers have a service time  $s_i = 0$ . The vertex sequence in the example is  $(0, v_1, v_2, v_3, v_4, v_5, 0)$  and is assumed to be capacity feasible. We set the battery capacity  $B$  to 7, so that a recharge is necessary. In the given sequence, recharging decision 1, corresponding to route  $(0, v_1, r_1, v_2, v_3, v_4, v_5, 0)$ , causes time-window violations of 1 unit at customer  $v_2$  and 2 units at customer  $v_4$ , implying  $f_{gen} = 9 + 3\beta$ . Recharging decision 2, corresponding to route  $(0, v_1, v_2, v_3, v_4, r_2, v_5, 0)$ , is time-window feasible and has a generalized objective value of  $f_{gen} = 12$ . Thus, if  $\beta < 1$ , decision 1 is chosen because the detour via  $r_2$  is more costly than the penalty for the time-window violations at customers  $v_2$  and  $v_4$ . Otherwise, if  $\beta > 1$ , recharging decision 2 leads to a smaller objective value.

### 3.2 | Penalty mechanism

Infeasible solutions with respect to vehicle capacity and time windows are evaluated based on the generalized cost function (Algorithm 3)

$$f_{gen}(\Sigma) = \sum_{\sigma \in \Sigma} f_{gen}(\sigma, \alpha, \beta),$$

where  $\alpha$  and  $\beta$  are the penalty factors for weighting the violations.

The LNS  $\times$  GTS heuristic dynamically updates the penalty factors based on the course of the search. Both penalty factors are initially set to  $\alpha = \beta = 1$  and are restricted to the interval  $[0.01, 100]$ . At every iteration, the respective penalty factor is multiplied (divided) by 1.1 in case the solution is infeasible (feasible) with regard to the respective resource. From then on, the update is repeated after every iteration until the feasibility status changes. The utilized parameter values are predetermined values, which are similar to those found in the literature for related problems. No fine tuning of this parameters was conducted, and we rely on the adaptive nature of this technique to guide the search into promising directions.

### 3.3 | LNS component

LNS was originally introduced by Shaw [34] and extended to ALNS by Ropke and Pisinger [30]. The idea is to iteratively destroy and repair larger parts of an initial solution to gradually improve the solution. (A)LNS has successfully been applied to many hard routing problems.

**Algorithm 3:** Generalized cost function  $f_{gen}(\sigma, \alpha, \beta)$  for a vertex sequence  $\sigma$ 


---

```

input : vertex sequence  $\sigma = (i_0, \dots, i_p)$ , penalty factors  $\alpha$  and  $\beta$ 
output: either INFEASIBLE or generalized cost
1  $b \leftarrow \sum_{j=1}^p b_{i_{j-1}i_j}$ 
2 if  $b > 2B$  then
3    $\lfloor$  return INFEASIBLE
4  $P_{cap} \leftarrow \max\{0, \sum_{j=1}^{p-1} d_{i_j} - Q\}$ 
5 if  $b > B$  then
6   /* recharging required, determine optimal recharging decisions */
7    $f^* \leftarrow \infty$ 
8   Compute  $j_{min}$  and  $j_{max}$  via Eq. (4)
9    $b^1 \leftarrow \sum_{k=1}^{j_{min}-1} b_{i_{k-1}i_k}, b^2 \leftarrow \sum_{k=j_{min}+1}^p b_{i_{k-1}i_k}$ 
10  for  $j \in [j_{min}, j_{max}]$  do
11    for  $r \in R(i_{j-1}, i_j)$  do
12      if  $b^1 + b_{i_{j-1},r} \leq B$  and  $b_{r,i_j} + b^2 \leq B$  then
13        /* route is battery-capacity feasible */
14         $\omega = (i_0^{\circ}, i_1^{\circ}, \dots, i_{p+1}^{\circ}) \leftarrow (i_0, \dots, i_{j-1}, r, i_j, \dots, i_p)$  // insert  $r$  and renumber
15        Compute  $(y_j)_{j=0}^{p+1}$  for route  $\omega$  via Eq. (1)
16        Compute  $P_{tw}$  for route  $\omega$  and service times  $(S_j) = (s_{i_j^{\circ}} + y_j)$ 
17         $f \leftarrow c_{\omega} + \alpha \cdot P_{cap} + \beta \cdot P_{tw}$ 
18        if  $f < f^*$  then
19           $\lfloor f^* \leftarrow f$ 
20     $b^1 \leftarrow b^1 + b_{j-1,j}, b^2 \leftarrow b^2 - b_{j-1,j}$ 
21  /* check battery-capacity feasibility again */
22  if  $f^* = \infty$  then
23     $\lfloor$  return INFEASIBLE
24  return  $f^*$ 
25 else
26   /* standard case as in VRPTW */
27   Compute  $P_{tw}$  for route  $\omega = \sigma$ 
28   return  $c_{\sigma} + \alpha \cdot P_{cap} + \beta \cdot P_{tw}$ 

```

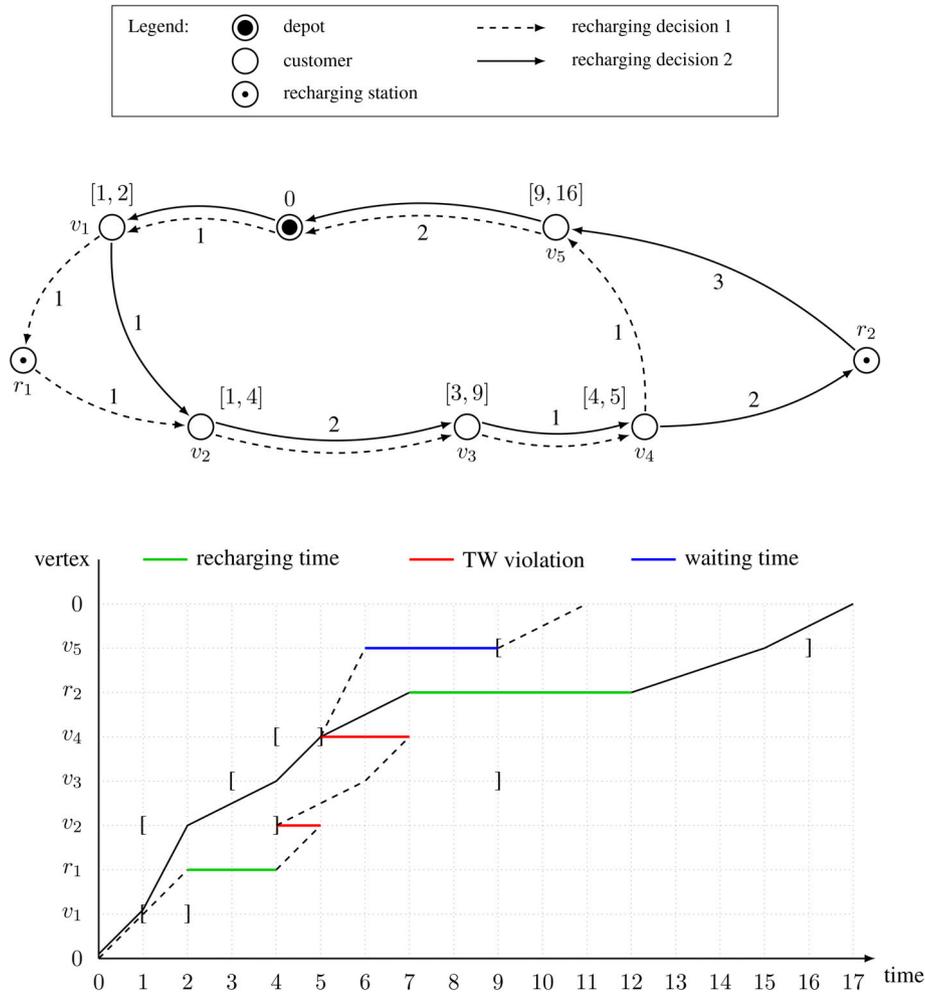
---

In our algorithm, the main purpose of the LNS step is diversification. First, we remove a randomly drawn number  $\delta$  of customers from the solution, where  $\delta$  is uniformly distributed over the interval  $[0.2n, 0.8n]$ . Contrary to many LNS implementations (see, e.g., [4,30]), we insert all removed customers in each LNS step. The resulting infeasible solutions are handled by the penalty mechanism described above (see also [3,13]). Due to the implicit recharging decision, no EVRPTWS-tailored operators for handling recharging visits are necessary. The search runs until no improving solution has been found for  $\eta_{lns} = 10$  consecutive iterations.

As repair operator, a combination of Regret-2 and GRASP (introduced in [10]), called Regret/GRASP repair, is used. Analogous to Regret-2 insertion (see, e.g. [30]), the difference in cost increase for the insertion of all remaining vertices at their best and at their second best position is calculated, and the vertices are stored in a list in descending order of this regret value. Instead of selecting the first vertex in the list for insertion, one of the first three vertices is randomly drawn with uniform probability.

### 3.4 | GTS component

At each main iteration, LNS  $\times$  GTS tries to improve the solution returned by the LNS by a GTS that is run for  $\eta_{gts} = 2000$  consecutive iterations. GTS uses a composite neighborhood of the operators 2-opt\*, relocate, split-0, and split-1. Moves are realized by means of the *generator-arc principle* introduced by Toth and Vigo [37]. Again, no specialized operators addressing recharging visits are necessary due to the implicit representation. Figure 2 illustrates the different neighborhood operators, where the respective generator arc is given by  $(v, w)$ .



**FIGURE 1** Vertex sequence (0,  $v_1, v_2, v_3, v_4, v_5, 0$ ), two possible associated routes with recharging, and resulting schedules with time-warp demonstrating the dependency of the implicit recharging decision on the penalty factor  $\beta$  [Color figure can be viewed at wileyonlinelibrary.com]

Relocate and 2-opt\* are well-known operators; relocate is applied for intra- and inter-route moves, and 2-opt\* is only defined for inter-route moves. The split-0 operator removes one arc from an existing vertex sequence and reconnects the loose ends to the depot; the split-1 operator creates a new vertex sequence containing a single customer. 2-opt\* and relocate are able to close a route if  $v_+$  and  $w_-$  or  $v_+$  and  $v_-$  are depot nodes, respectively, while both split operators always increase the number of routes by one. For a given solution, the 2-opt\* and relocate neighborhoods both contain  $\mathcal{O}(n^2)$  elements, whereas the split neighborhoods contain  $\mathcal{O}(n)$  elements.

Reinsertion of a removed arc is tabu for a random tenure out of the interval [5, 10] (based on Schneider et al. [33]).

We keep the granularity features of the GTS as simple as possible but make use of the insights on the design of granular solution methods for routing problems with time windows described by Schneider et al. [33]:

- As sparsification criterion to obtain a reduced list of possible generator arcs, we solve the network relaxation of VRPTW (see, e.g., [1]) using time-adjusted costs (TAC) as cost matrix. TAC are defined as  $c'_{ij} = \max\{e_j - (\ell_i + s_i); c_{ij}\}$ , thus favoring arcs with low travel cost and small waiting time. We solve the network relaxation by means of the Hungarian algorithm [23] and sort the arcs according to nondecreasing values of the reduced costs. In the base setting, we keep the 10% of arcs with the smallest reduced costs, that is, we use a sparsification threshold of 0.1.
- All arcs from the best solution found so far are added to the set of generator arcs.
- We use dynamic sparsification, that is, we adapt the strength of the sparsification to diversify and intensify the search: the sparsification threshold in the  $k$ th iteration since the last improvement is given by  $0.1 + 0.9 \cdot \left(\frac{k}{n_{gts}}\right)^2$ . As soon as an improving solution is found, we reset the sparsification threshold to 0.1.

GTS uses a continuous diversification mechanism similar to that of Cordeau et al. [2]. Let  $\Delta_A(\Sigma, \Sigma')$  denote the arcs which are not traversed in solution  $\Sigma$  but are traversed in solution  $\Sigma'$ . Moreover, let  $\Lambda(a)$  be a function that counts the number of times that an arc  $a \in A$  has been inserted into the incumbent solution by the GTS. For a nonimproving move that leads from solution  $\Sigma$

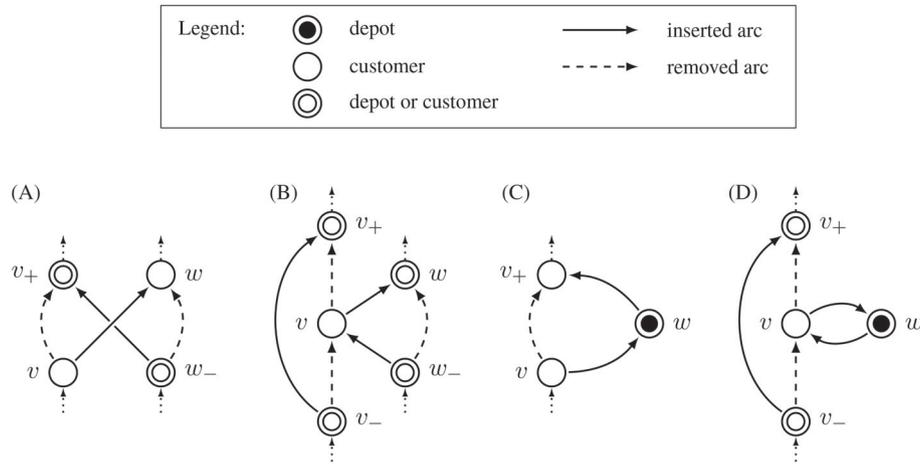


FIGURE 2 Neighborhood operators of GTS. A, 2-opt\*. B, relocate. C, split-0. D, split-1

to solution  $\Sigma'$ , that is, in case that  $f_{gen}(\Sigma') \geq f_{gen}(\Sigma)$ , we add a diversification penalty to the objective function value as follows:

$$f_{div}(\Sigma') = f_{gen}(\Sigma') + 10 \cdot f(\Sigma') \cdot \sum_{a \in \Delta_A(\Sigma, \Sigma')} \Lambda(a).$$

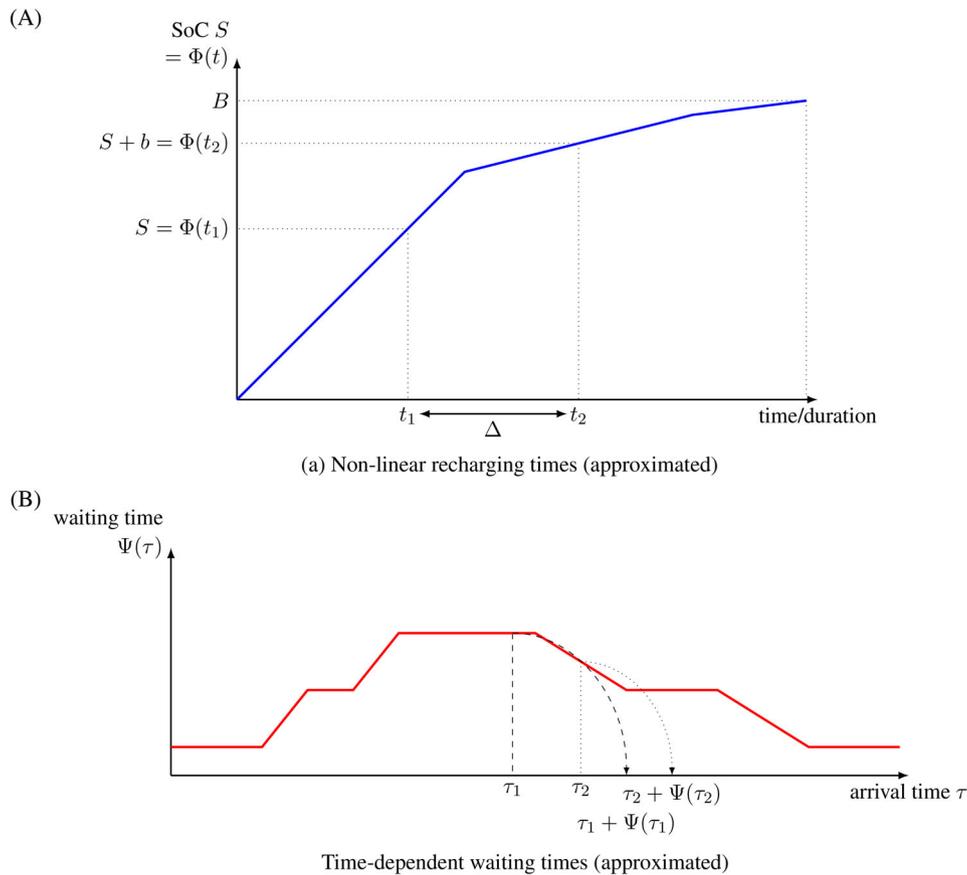
### 3.5 | Straightforward extensions

This section describes how some of the recently proposed extensions studied in the EVRP literature (see Section 1) could be incorporated into our LNS  $\times$  GTS algorithm to arrive at more detailed and then perhaps more realistic EVRPTWS variants. We briefly sketch three possible extensions—namely nonlinear charging times, chargers with different recharging rates, and time-dependent waiting times at recharging stations—and the necessary modifications of our algorithmic components.

First, our approach could easily incorporate *nonlinear charging times*, that is, recharging times that depend of the current *state of charge* (SoC) as considered by Montoya et al. [22] and Froger et al. [11]. For this extension, we have to strictly distinguish between energy and time/duration so that a direct time-energy correspondence using Equation (1) is no longer possible. The nonlinear charging times in the cited works are represented by a monotone function  $\Phi: \mathbb{R}_+ \rightarrow [0, B]$  that models the interdependency between an SoC  $S$  and the duration  $\Delta$  of the charging operation as depicted in Figure 3A. Given the SoC  $S$  when arriving at a recharging station  $r \in R$  and a duration  $\Delta$  to spend on recharging, the value  $\Phi(\Phi^{-1}(S) + \Delta)$  is the resulting SoC when departing from the recharging station. Conversely, to reach the SoC  $S + b$  for  $b \leq B - S$ , a loading time of  $\Delta = \Phi^{-1}(S + b) - \Phi^{-1}(S)$  time units is required. Hence, the only required modifications in Algorithm 3 are that in Step 13 the loading plan must be computed with a modified Equation (1) that uses the inverse function  $\Phi^{-1}$ . The essential point is that with a single recharge, the amount consumed before reaching the recharging station (and therefore also the SoC) as well as the amount that must be recharged is known in Step 13. In Step 14, we then set  $(S_j) = (s_j^e + \Delta_j)$ , where  $\Delta_j$  is the new computed recharging duration.

Second, the incorporation of heterogeneous stations with chargers having *different recharging rates* (e.g., [8]) is possible by using station-specific functions  $\Phi_r$  for  $r \in R$  instead of a single recharging function.

Third, it is also straightforward to consider *time-dependent waiting times* at recharging stations as suggested by Keskin et al. [19]. The station-specific function  $\Psi_r(\tau)$  defined over possible arrival times  $\tau$  models that, due to congestion at recharging station  $r$ , an ECV arriving at time  $\tau$  must wait there for an (expected) waiting time of  $\Psi_r(\tau)$ . Keskin et al. [19] derive an approximate function from expected time-dependent arrival rates and the corresponding expected waiting times resulting from a queueing model. A key assumption is that the FIFO property  $\tau_1 + \Psi_r(\tau_1) \leq \tau_2 + \Psi_r(\tau_2)$  for all stations  $r \in R$  and arrival times with  $\tau_1 \leq \tau_2$  holds, that is, arriving later at recharging station  $r$  never leads to an earlier departure, see Figure 3B. A consequence of this assumption is that an as-early-as-possible schedule, as used in our Algorithm 2, leads to minimal time-window violations and penalties  $P_{tw}$  (if any).



**FIGURE 3** Possible extensions. (A) Nonlinear recharging times (approximated). (B) Time-dependent waiting times (approximated) [Color figure can be viewed at [wileyonlinelibrary.com](http://wileyonlinelibrary.com)]

## 4 | COMPUTATIONAL EXPERIMENTS AND RESULTS

In this section, we report our computational results on  $\text{LNS} \times \text{GTS}$  for the EVRPTWS. All results were obtained using a computing cluster running CentOS 7.6 with Intel Xeon E5-2430v2 processors clocked at 3.0 GHz, where 1 GB of RAM was available to a single process. The  $\text{LNS} \times \text{GTS}$  algorithm was implemented in C++ and compiled into 64-bit single-thread release code with GCC 9.1.

We would like to stress that the quality of our approach does not depend on an extensive fine-tuning of parameters. Indeed, parameters of the algorithm are set according to values from the literature for similar algorithmic components (as described above) or based on a rough tuning during the development of the algorithm. Therefore, we do not present a study on choosing parameter values for  $\text{LNS} \times \text{GTS}$ .

The remainder of this section is structured as follows: The benchmark instances from the literature are described in Section 4.1. The results of a performance analysis of  $\text{LNS} \times \text{GTS}$  on EVRPTWS are reported in Section 4.2. We investigate the cost savings potential of partial recharges in comparison to full recharges in the presence of time-window constraints in Section 4.3. On the basis of a systematic examination, we highlight factors that influence this cost saving potential.

### 4.1 | Benchmark instances

The performance of  $\text{LNS} \times \text{GTS}$  is investigated on the EVRPTWS benchmark instances of Desaulniers et al. [5]. They were derived from the EVRPTW benchmark set with multiple recharges of Schneider et al. [32], which in turn is based on the original VRPTW instances from Solomon [35]. The instance set features instances with 25, 50, and 100 customers, which are geographically distributed in three different ways: clustered distribution (C), uniform random distribution (R), and a mixture of both random and clustered (RC). Furthermore, instances C1, R1, and RC1 (type 1) have a shorter scheduling horizon and smaller vehicle capacities compared to instances C2, R2, and RC2 (type 2). As a consequence, type 1 instances require a larger number of routes compared to those of type 2. Recharging times  $b_{ij}$  depend linearly on the amount of energy that is recharged, which is in turn proportional to the traveled distance  $c_{ij}$ . All EVRPTWS test instances are available as supplemental download at INFORMS, doi: 10.1287/opre.2016.1535.

**TABLE 1** Performance of LNS  $\times$  GTS on the EVRPTWS benchmark of Desaulniers et al. [5] grouped by instance classes, also compared against results obtained with a plain GTS run for  $\eta_{gts} = 25\,000$  iterations and a plain LNS run for  $\eta_{lms} = 1500$  iterations

Instance group	$ N  = 25$			$ N  = 50$			$ N  = 100$		
	$\Delta_b$	$\Delta_a$	$t$	$\Delta_b$	$\Delta_a$	$t$	$\Delta_b$	$\Delta_a$	$t$
<i>Full recharge (F)</i>									
C1	0.00	0.00	9.28	0.00	0.00	48.92	0.06	0.10	223.50
C2	0.00	0.00	8.77	0.00	0.00	49.97	0.00	0.00	242.21
R1	0.00	0.00	9.80	0.00	0.03	68.03	0.25	0.67	350.90
R2	0.00	0.00	11.78	0.00	0.06	96.01	0.02	0.17	676.15
RC1	0.00	0.00	9.26	0.01	0.20	62.94	0.29	0.68	357.92
RC2	0.00	0.00	11.51	0.00	0.00	74.93	0.05	0.11	507.52
All	0.00	0.00	10.12	0.00	0.05	68.13	0.12	0.31	402.16
GTS only	0.00	0.00	13.71	0.00	0.08	75.09	0.26	0.72	397.19
LNS only	0.01	0.18	4.44	1.19	2.24	43.66	3.76	5.11	390.31
<i>Partial recharge (P)</i>									
C1	0.00	0.00	8.52	0.00	0.01	49.68	0.37	0.43	229.17
C2	0.00	0.00	8.79	0.00	0.00	50.02	0.00	0.00	240.96
R1	0.00	0.00	9.63	0.09	0.12	63.91	0.29	0.64	355.20
R2	0.00	0.00	11.70	0.00	0.05	92.45	0.05	0.18	704.55
RC1	0.00	0.00	9.59	0.02	0.22	74.42	0.26	0.59	377.03
RC2	0.00	0.00	11.79	0.00	0.00	76.29	0.00	0.06	512.40
All	0.00	0.00	10.04	0.02	0.07	68.51	0.17	0.33	412.82
GTS only	0.00	0.00	13.74	0.02	0.06	79.46	0.24	0.65	405.73
LNS only	0.01	0.22	4.55	1.07	2.08	49.21	3.87	5.22	427.66

## 4.2 | Performance of LNS $\times$ GTS

We first analyze the performance of LNS  $\times$  GTS on the basis of the best-known solutions (BKS), which are given by the best of the solutions found either by Desaulniers et al. [5,6], or LNS  $\times$  GTS during the complete course of its development. Aggregated results are given in Table 1, where instances are grouped by size, recharge type, and customer distribution. For each instance, we conducted five runs of LNS  $\times$  GTS. Columns  $\Delta_b$  and  $\Delta_a$  show the best and average gap to the BKS over the five runs, respectively. Column  $t$  denotes the average runtime over five runs in seconds. In the Appendix, the same results are given on an instance basis in Tables A1 to A3. In addition, Table A4 shows the results of LNS  $\times$  GTS compared to the previous BKS (i.e., the best of the results of Desaulniers et al. [5,6]) on the instances with 100 customers.

The aggregated results in Table 1 show that LNS  $\times$  GTS finds the optimal solution of each of the 25-customer instances in each individual run. For the instances with 50 and 100 customers, the gaps of LNS  $\times$  GTS stay quite low, and no significant difference can be found comparing the partial and full recharge policies. In addition, LNS  $\times$  GTS shows a robust performance, that is, the difference between best and average solution quality is relatively small. Due to the implicit representation of recharges used by the algorithm, computation times increase significantly with instance sizes. However, an average runtime of less than 7 minutes for instances with 100 customers remains adequate for most application scenarios. We find that LNS  $\times$  GTS on average matches 222 of the previous BKS provided by Desaulniers et al. [5,6], and it matches 253 of the 328 when considering the best out of five runs. Further, with the best out of five runs, LNS  $\times$  GTS is able to improve the previous BKS on 35 instances, 33 of which are 100-customer instances (see Table A4). The amount of improvement is substantial on many of the large instances, resulting in average gaps of  $-1.13\%$  and  $-0.81\%$  for the F and P policies, respectively.

Table 1 also includes a brief comparison of LNS  $\times$  GTS against a plain GTS and a plain LNS heuristics (rows “GTS only” and “LNS only”). In order to arrive at comparable computations times, in particular for the 100-customer instances,  $\eta_{gts} = 25\,000$  iterations are allowed in the GTS and  $\eta_{lms} = 1500$  iterations in the LNS. The results for the instances with  $|N| = 50$  and 100 customers clearly prove that the combination of GTS and LNS is beneficial.

## 4.3 | Cost benefits of the partial recharge option

We investigate the cost benefits of allowing partial recharges (P) instead of forcing full recharges (F), and the influence of

1. the number and geographical distribution of customers (Section 4.3.1),
2. the time-window density (Section 4.3.2), and
3. the time-window width (Section 4.3.3).

**TABLE 2** Change in objective value (in percent) when allowing P compared to F on the EVRPTWS instances grouped by customer distribution and size

N	Instance group						All
	C1	C2	R1	R2	RC1	RC2	
25	-1.33	0.00	-0.88	0.00	-0.48	0.00	-0.47
50	-1.37	0.00	-1.17	-0.02	-1.16	0.00	-0.64
100	-1.19	-1.99	-1.49	-0.02	-1.36	0.00	-0.99
All	-1.29	-0.66	-1.18	-0.01	-1.00	0.00	-0.70

**TABLE 3** Change in objective value (in percent) obtained by LNS  $\times$  GTS when allowing partial recharges compared to only full recharges on EVRPTWS instances with varying time-window densities

N	Time-window density					
	0%	20%	40%	60%	80%	100%
25	-0.10	-0.24	-0.28	-0.27	-0.19	-0.91
50	-0.16	-0.07	-0.25	-0.54	-0.38	-0.53
100	-0.22	-0.54	-0.76	-1.00	-0.92	-0.89
All	-0.16	-0.28	-0.43	-0.60	-0.50	-0.78

#### 4.3.1 | Influence of problem size and geographical customer distribution

We base our comparison between partial and full recharges on the BKS for each individual instance and recharge type. To analyze the influence of the problem size and geographical distribution of customers on the BKS, Table 2 groups the instances according to customer distribution type and instance size |N|. The values in the table report the average decrease in routing cost when allowing partial recharges (P) compared to requiring full recharges (F) in percent.

We first observe that the improvement in objective function value by allowing P on type 2 instances is negligible except for the instances of group C2 with 100 customers. On these particular instances, we found that with the F policy, the time required for the full recharge and the time-window constraints prevent the routes from naturally following the given clusters, which leads to an average detour of almost 2% compared to the P policy. For type 1 instances, we observe an average improvement of the objective value of 1.16% by allowing P. This result is consistent with the observations of Desaulniers et al. [5], who report an average improvement of 0.97% for those instances of type 1 for which a feasible solution was found by their exact approach. We find that the benefit is correlated to the instance size: for 25 customers, the benefit amounts to approximately 0.5%, whereas for 50 and 100 customers, we achieve benefits of approximately 0.65% and 1%, respectively. Indeed, with more customers, the routes are more productive, that is, there are more customers per route and less waiting time before the opening of the time windows. Consequently, partial recharges become more useful.

#### 4.3.2 | Influence of time-window density

We next evaluate the influence of different *time-window densities* (TWDs), where the TWD of instances is the percentage of customers that have a constraining time window. For this purpose, we modify existing instances to get additional test instances with different TWDs. In order to eliminate effects caused by different customer distribution, we restrict ourselves to instances of classes R1 and R2 with a TWD of 100%. These are the 11 instances R101, R105, R109, R110, R111, R112, R201, R205, R206, R207, and R208. From each of these 33 base instances (11 instances with 25, with 50, and with 100 customers), we generate five new instances with a TWD of 80%, 60%, 40%, 20%, 0%, respectively, by consecutively removing customer time windows in 20% steps in random fashion. Aggregated results are given in Table 3. The reported values are the routing cost benefits of policy P compared to F in percent, based on the BKS for each instance.

We can observe the rough tendency that the benefit of partial recharges increases with TWD. The relationship is not entirely consistent and the effect is rather small.

Moreover, without any time windows (a TWD of 0%), one would expect that there is no difference between the P and F solutions. However, we observe a small difference of -0.16%. Two factors may explain this difference: First, the overall time horizon still constrains the duration of the routes, stronger for the policy F than for P. Second, the LNS  $\times$  GTS heuristic may perform slightly better for the P policy. The definite reason for the difference remains unclear.

**TABLE 4** Change in objective value obtained by LNS  $\times$  GTS when allowing P compared to F on EVRPTWS instances with varying time-window widths

N	Time-window width				
	$\varphi = 1$	$\varphi = 2$	$\varphi = 3$	$\varphi = 5$	$\varphi = 10$
25	-1.66	-3.68	-1.96	-0.70	-0.18
50	-0.93	-1.16	-0.72	-0.27	-0.32
100	-1.60	-0.96	-0.85	-0.82	-0.47
All	-1.40	-1.93	-1.18	-0.60	-0.32

### 4.3.3 | Influence of time-window width

To analyze the effect of time-window width, we use the EVRPTWS R1 instances with a 100% TWD as base instances. From each of these 6 instances, we generate four new instances by multiplying the *time-window width* (TWW) by a factor  $\varphi \in \{2, 3, 5, 10\}$ . More precisely, if a customer originally has a time window beginning at  $a$  and ending at  $b$  of width  $w = b - a$  centered around  $c = 0.5 \cdot (a + b)$ , then the customer has a time window beginning at  $a' = c - 0.5 \cdot \varphi \cdot w$  and ending at  $b' = c + 0.5 \cdot \varphi \cdot w$  on the modified instance. Thus, the TWW given by  $\varphi = 1$  corresponds to the original instance. The planning horizon, that is, the time window of the depot remains unchanged. Table 4 reports the routing cost benefit of the P policy compared to the F policy in percent, again based on the BKS.

We observe the clear tendency that the benefit of partial recharges decreases with wider customer time windows although the relationship is not entirely consistent, that is, for the instance classes with different sizes it happens frequently that an increase of TWW also leads to an increase in benefits of the partial recharge. However, recall that the latter is a relative measure that relates the costs of the P and F policies.

## 5 | CONCLUSION

This paper presents a hybrid of LNS and GTS, called LNS  $\times$  GTS, to address the EVRPTW with single recharge and partial as well as full recharge. LNS  $\times$  GTS handles the recharging decisions implicitly, that is, the search operators work on customer sequences and the recharging is always carried out in an optimal fashion. LNS  $\times$  GTS shows a convincing performance on EVRPTWS instances from the literature. The computational experiments also reveal that allowing partial recharges allows us to notably reduce routing cost, an effect that increases with growing instance size. In addition, we find that partial recharges tend to be more beneficial compared to full recharges if many customers have time-window requirements and if the customer time windows are narrow.

### ACKNOWLEDGMENT

This research was partially supported by Deutsche Forschungsgemeinschaft (DFG) under grants IR122/7-1 and SCHN 1497/1-1.

### ORCID

Stefan Irnich  <https://orcid.org/0000-0001-9383-4546>

### REFERENCES

- [1] J.-F. Cordeau, G. Desaulniers, J. Desrosiers, M. Solomon, and F. Soumis. "VRP with time windows", P. Toth and D. Vigo (eds.), *The Vehicle Routing Problem, Monographs on Discrete Mathematics and Applications, Chapter 7*, Philadelphia, PA: SIAM, 2002, pp. 157–193.
- [2] J.-F. Cordeau, M. Gendreau, and G. Laporte. *A tabu search heuristic for periodic and multi-depot vehicle routing problems*, *Networks* **30** (1997), 105–119.
- [3] I. Dayarian, T.G. Crainic, M. Gendreau, and W. Rei. *An adaptive large-neighborhood search heuristic for a multi-period vehicle routing problem*, *Transp. Res. E* **95** (2016), 95–123.
- [4] E. Demir, T. Bektaş, and G. Laporte. *An adaptive large neighborhood search heuristic for the pollution-routing problem*, *Eur. J. Oper. Res.* **223** (2012), 346–359.
- [5] G. Desaulniers, F. Errico, S. Irnich, and M. Schneider. *Exact algorithms for electric vehicle-routing problems with time windows*, *Oper. Res.* **64** (2016), 1388–1405.
- [6] G. Desaulniers, T. Gschwind, and S. Irnich. *Variable fixing for two-arc sequences in branch-price-and-cut algorithms on path-based models*, *Les Cahiers du GERAD G-2019-48*, GERAD, Montréal, Canada, 2019.
- [7] T. Erdelić and T. Carić. *A survey on the electric vehicle routing problem: Variants and solution approaches*, *J. Adv. Transp.* **2019** (2019), 1–48.

- [8] Á. Felipe, M.T. Ortuño, G. Righini, and G. Tirado, *A heuristic approach for the green vehicle routing problem with multiple technologies and partial recharges*, *Transp. Res. E* **71** (2014), 111–128.
- [9] W. Feng and M.A. Figliozzi, *An economic and technological analysis of the key factors affecting the competitiveness of electric commercial vehicles: A case study from the USA market*, *Transp. Res. C* **26** (2013), 135–145.
- [10] T.A. Feo and M.G.C. Resende, *A probabilistic heuristic for a computationally difficult set covering problem*, *Oper. Res. Lett.* **8** (1989), 67–71.
- [11] A. Froger, J.E. Mendoza, O. Jabali, and G. Laporte, *Improved formulations and algorithmic components for the electric vehicle routing problem with nonlinear charging functions*, *Comput. Oper. Res.* **104** (2019), 256–294.
- [12] A. Froger, J.E. Mendoza, O. Jabali, and G. Laporte, *The electric vehicle routing problem with capacitated charging stations*, Working paper, 2019b. <https://hal.archives-ouvertes.fr/hal-02386167>.
- [13] V.C. Hemmelmayr, J.-F. Cordeau, and T.G. Crainic, *An adaptive large neighborhood search heuristic for two-echelon vehicle routing problems arising in city logistics*, *Comput. Oper. Res.* **39** (2012), 3215–3228.
- [14] V.C. Hemmelmayr, K.F. Doerner, R.F. Hartl, and S. Rath, *A heuristic solution method for node routing based solid waste collection problems*, *J. Heurist.* **19** (2013), 129–156.
- [15] G. Hiermann, J. Puchinger, S. Ropke, and R.F. Hartl, *The electric fleet size and mix vehicle routing problem with time windows and recharging stations*, *Eur. J. Oper. Res.* **252** (2016), 995–1018.
- [16] G. Hiermann, R.F. Hartl, J. Puchinger, and T. Vidal, *Routing a mix of conventional, plug-in hybrid, and electric vehicles*, *Eur. J. Oper. Res.* **272** (2019), 235–248.
- [17] International Energy Agency, 2020. Electric vehicles initiative. <https://www.iea.org/programmes/electric-vehicles-initiative> (Accessed March 6, 2020).
- [18] M. Keskin and B. Çatay, *Partial recharge strategies for the electric vehicle routing problem with time windows*, *Transp. Res. C* **65** (2016), 111–127.
- [19] M. Keskin, G. Laporte, and B. Çatay, *Electric vehicle routing problem with time-dependent waiting times at recharging stations*, *Comput. Oper. Res.* **107** (2019), 77–94.
- [20] P.R. Kleindorfer, A. Neboian, A. Roset, and S. Spinler, *Fleet renewal with electric vehicles at La Poste*, *Interfaces* **42** (2012), 465–477.
- [21] A. Montoya, C. Guéret, J.E. Mendoza, and J.G. Villegas, *A multi-space sampling heuristic for the green vehicle routing problem*, *Transp. Res. C* **70** (2016), 113–128.
- [22] A. Montoya, C. Guéret, J.E. Mendoza, and J.G. Villegas, *The electric vehicle routing problem with nonlinear charging function*, *Transp. Res. B* **103** (2017), 87–110.
- [23] J. Munkres, *Algorithms for the assignment and transportation problems*, *J. Soc. Ind. Appl. Math.* **5** (1957), 32–38.
- [24] National Renewable Energy Laboratory, 2014. Project startup: Evaluating the performance of Frito Lay’s electric delivery trucks. <http://www.nrel.gov/docs/fy14osti/61455.pdf> (Accessed March 6, 2020).
- [25] S. Pelletier, O. Jabali, and G. Laporte, *50th anniversary invited article—Goods distribution with electric vehicles: Review and research perspectives*, *Transp. Sci.* **50** (2016), 3–22.
- [26] S. Pelletier, O. Jabali, G. Laporte, and M. Veneroni, *Battery degradation and behaviour for electric vehicles: Review and numerical analyses of several models*, *Transp. Res. B* **103** (2017), 158–187.
- [27] S. Pelletier, O. Jabali, and G. Laporte, *Charge scheduling for electric freight vehicles*, *Transp. Res. B* **115** (2018), 246–269.
- [28] S. Pelletier, O. Jabali, and G. Laporte, *The electric vehicle routing problem with energy consumption uncertainty*, *Transp. Res. B* **126** (2019), 225–255.
- [29] C. Prins, *A simple and effective evolutionary algorithm for the vehicle routing problem*, *Comput. Oper. Res.* **31** (2004), 1985–2002.
- [30] S. Ropke and D. Pisinger, *An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows*, *Transp. Sci.* **40** (2006), 455–472.
- [31] M. Schiffer, M. Schneider, G. Walther, and G. Laporte, *Vehicle routing and location-routing with intermediate stops: A review*, *Transp. Sci.* **53** (2019), 319–622.
- [32] M. Schneider, A. Stenger, and D. Goeke, *The electric vehicle-routing problem with time windows and recharging stations*, *Transp. Sci.* **48** (2014), 500–520.
- [33] M. Schneider, F. Schwahn, and D. Vigo, *Designing granular solution methods for routing problems with time windows*, *Eur. J. Oper. Res.* **2** (2017), 493–509.
- [34] P. Shaw, “Using constraint programming and local search methods to solve vehicle routing problems”. M. Maher and J.-F. Puget (eds.), *Principles and Practice of Constraint Programming – CP98, Lecture Notes in Computer Science*, vol. 1520, Springer, London, UK, 1998, pp. 417–431.
- [35] M.M. Solomon, *Algorithms for the vehicle routing and scheduling problems with time window constraints*, *Oper. Res.* **35** (1987), 254–265.
- [36] Tesla Motors, Inc. Supercharger, 2020. <http://www.teslamotors.com/supercharger> (Accessed March 6, 2020).
- [37] P. Toth and D. Vigo, *The granular tabu search and its application to the vehicle-routing problem*, *INFORMS J. Comput.* **15** (2003), 333–346.
- [38] P. Toth and D. Vigo (eds), *Vehicle Routing: Problems, Methods, and Applications*, *MOS-SIAM Series on Optimization*, vol. 18, Philadelphia: SIAM, 2014.
- [39] J. Van Duin, L. Tavasszy, and H. Quak, *Towards E(lectric)-urban freight: First promising steps in the electric vehicle revolution*, *Eur. Transp.* **54** (2013), 1–19.
- [40] T. Vidal, T.G. Crainic, M. Gendreau, and C. Prins, *A hybrid genetic algorithm with adaptive diversity management for a large class of vehicle routing problems with time-windows*, *Comput. Oper. Res.* **40** (2013), 475–489.

**How to cite this article:** Löffler M, Desaulniers G, Irnich S, Schneider M. Routing electric vehicles with a single recharge per route. *Networks*. 2020;76:187–205. <https://doi.org/10.1002/net.21964>

## APPENDIX A

We provide computational results for each individual EVRPTWS instance. Results are grouped by instance size (25, 50, and 100) in Tables A1–A3. The following information is given:

Instance	Name of the problem instance
BKS	Best known solution
Underlined	Optimality proven by either [5] or [6]
$\Delta_b$	Best gap relative to BKS in percent over five runs
$\Delta_a$	Average gap relative to BKS in percent over five runs
$t$	Average computation time over five runs in seconds

Note that the BKS values are given by the best of the solutions found by Desaulniers et al. [5,6], and LNS  $\times$  GTS during the complete course of its development.

Table A4 compares the performance of LNS  $\times$  GTS on the 100-customer instances to the previous BKS, that is, only the solutions from Desaulniers et al. [5,6] are considered.

TABLE A1 Results of LNS  $\times$  GTS on the 25-customer EVRPTW-S instances

Instance	Full recharges (F)				Partial recharges (P)			
	BKS	$\Delta_b$	$\Delta_a$	$t$	BKS	$\Delta_b$	$\Delta_a$	$t$
C101	<u>642.58</u>	0.00	0.00	7.00	<u>622.89</u>	0.00	0.00	6.63
C102	<u>543.78</u>	0.00	0.00	12.02	<u>541.30</u>	0.00	0.00	11.71
C103	<u>345.54</u>	0.00	0.00	8.81	<u>345.54</u>	0.00	0.00	8.95
C104	<u>459.74</u>	0.00	0.00	8.48	<u>459.68</u>	0.00	0.00	8.70
C105	<u>541.49</u>	0.00	0.00	5.74	<u>537.06</u>	0.00	0.00	6.08
C106	<u>564.60</u>	0.00	0.00	6.09	<u>557.65</u>	0.00	0.00	6.48
C107	<u>505.86</u>	0.00	0.00	6.70	<u>505.15</u>	0.00	0.00	6.81
C108	<u>531.64</u>	0.00	0.00	12.34	<u>509.15</u>	0.00	0.00	10.12
C109	<u>503.87</u>	0.00	0.00	16.36	<u>493.85</u>	0.00	0.00	11.16
C201	<u>416.37</u>	0.00	0.00	6.98	<u>416.37</u>	0.00	0.00	6.76
C202	<u>404.57</u>	0.00	0.00	10.82	<u>404.57</u>	0.00	0.00	10.58
C203	<u>391.72</u>	0.00	0.00	9.87	<u>391.72</u>	0.00	0.00	10.19
C204	<u>362.93</u>	0.00	0.00	11.01	<u>362.93</u>	0.00	0.00	11.20
C205	<u>409.05</u>	0.00	0.00	7.68	<u>409.05</u>	0.00	0.00	7.79
C206	<u>389.19</u>	0.00	0.00	7.17	<u>389.19</u>	0.00	0.00	7.15
C207	<u>400.37</u>	0.00	0.00	8.28	<u>400.37</u>	0.00	0.00	8.28
C208	<u>419.19</u>	0.00	0.00	8.34	<u>419.19</u>	0.00	0.00	8.33
R101	<u>672.41</u>	0.00	0.00	9.70	<u>662.30</u>	0.00	0.00	5.31
R102	<u>455.58</u>	0.00	0.00	6.95	<u>453.50</u>	0.00	0.00	8.13
R103	<u>499.23</u>	0.00	0.00	8.98	<u>498.58</u>	0.00	0.00	8.88
R104	<u>352.16</u>	0.00	0.00	9.26	<u>352.16</u>	0.00	0.00	9.40
R105	<u>621.89</u>	0.00	0.00	10.08	<u>616.38</u>	0.00	0.00	8.87
R106	<u>480.22</u>	0.00	0.00	8.83	<u>480.22</u>	0.00	0.00	10.01
R107	<u>416.51</u>	0.00	0.00	11.73	<u>416.51</u>	0.00	0.00	12.10
R108	<u>456.50</u>	0.00	0.00	6.37	<u>456.50</u>	0.00	0.00	6.71
R109	<u>467.68</u>	0.00	0.00	10.62	<u>462.03</u>	0.00	0.00	12.59
R110	<u>424.38</u>	0.00	0.00	13.73	<u>410.04</u>	0.00	0.00	14.88
R111	<u>383.04</u>	0.00	0.00	10.84	<u>383.04</u>	0.00	0.00	10.20
R112	<u>397.42</u>	0.00	0.00	10.45	<u>385.51</u>	0.00	0.00	8.48
R201	<u>374.28</u>	0.00	0.00	5.06	<u>374.28</u>	0.00	0.00	5.10
R202	<u>375.30</u>	0.00	0.00	6.96	<u>375.30</u>	0.00	0.00	6.43
R203	<u>382.77</u>	0.00	0.00	16.09	<u>382.77</u>	0.00	0.00	16.24
R204	<u>315.52</u>	0.00	0.00	14.15	<u>315.52</u>	0.00	0.00	13.93
R205	<u>374.00</u>	0.00	0.00	6.02	<u>374.00</u>	0.00	0.00	6.77
R206	<u>424.51</u>	0.00	0.00	14.82	<u>424.51</u>	0.00	0.00	15.25
R207	<u>330.96</u>	0.00	0.00	11.77	<u>330.96</u>	0.00	0.00	11.82

TABLE A1 Continued

Instance	Full recharges (F)				Partial recharges (P)			
	BKS	$\Delta_b$	$\Delta_a$	$t$	BKS	$\Delta_b$	$\Delta_a$	$t$
R208	<u>315.59</u>	0.00	0.00	17.41	<u>315.59</u>	0.00	0.00	17.55
R209	<u>422.39</u>	0.00	0.00	10.73	<u>422.39</u>	0.00	0.00	10.58
R210	<u>315.27</u>	0.00	0.00	12.59	<u>315.27</u>	0.00	0.00	12.48
R211	<u>383.81</u>	0.00	0.00	13.98	<u>383.81</u>	0.00	0.00	12.60
RC101	<u>763.35</u>	0.00	0.00	8.82	<u>763.35</u>	0.00	0.00	9.18
RC102	<u>675.66</u>	0.00	0.00	8.50	<u>675.66</u>	0.00	0.00	8.59
RC103	<u>563.83</u>	0.00	0.00	10.33	<u>552.23</u>	0.00	0.00	9.33
RC104	<u>519.20</u>	0.00	0.00	8.55	<u>519.20</u>	0.00	0.00	9.52
RC105	<u>594.69</u>	0.00	0.00	8.82	<u>594.69</u>	0.00	0.00	9.39
RC106	<u>557.23</u>	0.00	0.00	8.13	<u>557.23</u>	0.00	0.00	8.70
RC107	<u>502.22</u>	0.00	0.00	8.49	<u>502.22</u>	0.00	0.00	10.22
RC108	<u>479.73</u>	0.00	0.00	12.45	<u>471.08</u>	0.00	0.00	11.77
RC201	<u>594.89</u>	0.00	0.00	6.49	<u>594.89</u>	0.00	0.00	6.56
RC202	<u>540.87</u>	0.00	0.00	5.58	<u>540.87</u>	0.00	0.00	5.74
RC203	<u>443.28</u>	0.00	0.00	15.48	<u>443.28</u>	0.00	0.00	15.86
RC204	<u>445.51</u>	0.00	0.00	15.76	<u>445.51</u>	0.00	0.00	15.72
RC205	<u>536.35</u>	0.00	0.00	8.72	<u>536.35</u>	0.00	0.00	8.72
RC206	<u>589.19</u>	0.00	0.00	9.31	<u>589.19</u>	0.00	0.00	9.29
RC207	<u>485.88</u>	0.00	0.00	14.74	<u>485.88</u>	0.00	0.00	15.93
RC208	<u>440.24</u>	0.00	0.00	15.96	<u>440.24</u>	0.00	0.00	16.52

TABLE A2 Results of LNS  $\times$  GTS on the 50-customer EVRPTW-S instances

Instance	Full recharges (F)				Partial recharges (P)			
	BKS	$\Delta_b$	$\Delta_a$	$t$	BKS	$\Delta_b$	$\Delta_a$	$t$
C101	<u>783.90</u>	0.00	0.00	27.91	<u>776.93</u>	0.00	0.00	28.32
C102	<u>828.40</u>	0.00	0.00	60.90	<u>823.33</u>	0.00	0.00	50.21
C103	<u>668.33</u>	0.00	0.00	53.23	<u>667.60</u>	0.00	0.00	70.14
C104	<u>598.87</u>	0.00	0.00	97.89	<u>559.13</u>	0.00	0.00	77.86
C105	<u>757.11</u>	0.00	0.00	28.89	<u>754.74</u>	0.00	0.09	35.61
C106	<u>801.27</u>	0.00	0.00	35.18	<u>799.24</u>	0.00	0.00	36.77
C107	<u>792.96</u>	0.00	0.00	44.03	<u>782.90</u>	0.00	0.00	35.04
C108	<u>733.83</u>	0.00	0.00	45.27	<u>733.68</u>	0.00	0.00	48.59
C109	<u>690.19</u>	0.00	0.00	46.96	<u>675.07</u>	0.00	0.00	64.63
C201	<u>503.26</u>	0.00	0.00	34.61	<u>503.26</u>	0.00	0.00	34.98
C202	<u>523.75</u>	0.00	0.00	50.43	<u>523.75</u>	0.00	0.00	50.32
C203	<u>513.82</u>	0.00	0.00	60.56	<u>513.82</u>	0.00	0.00	59.83
C204	<u>469.56</u>	0.00	0.00	75.23	<u>469.56</u>	0.00	0.00	77.04
C205	<u>513.75</u>	0.00	0.00	38.91	<u>513.75</u>	0.00	0.00	39.53
C206	<u>515.97</u>	0.00	0.00	40.79	<u>515.97</u>	0.00	0.00	41.07
C207	<u>531.13</u>	0.00	0.00	51.80	<u>531.13</u>	0.00	0.00	50.55
C208	<u>528.40</u>	0.00	0.00	47.47	<u>528.40</u>	0.00	0.00	46.86
R101	<u>947.32</u>	0.00	0.00	37.17	<u>937.29</u>	0.00	0.00	38.90
R102	<u>909.72</u>	0.00	0.00	35.44	<u>878.39</u>	0.00	0.00	43.69
R103	<u>812.90</u>	0.00	0.00	41.28	<u>806.53</u>	0.00	0.00	58.41
R104	<u>624.14</u>	0.00	0.00	101.55	<u>615.55</u>	0.00	0.17	97.11
R105	<u>842.75</u>	0.00	0.00	35.63	<u>838.98</u>	0.00	0.00	39.61
R106	<u>811.47</u>	0.00	0.00	77.01	<u>799.38</u>	0.00	0.00	72.64
R107	<u>739.15</u>	0.00	0.00	71.79	<u>737.63</u>	0.21	0.21	61.23
R108	<u>543.83</u>	0.00	0.00	56.50	<u>537.63</u>	0.00	0.00	52.65

TABLE A2 Continued

Instance	Full recharges (F)				Partial recharges (P)			
	BKS	$\Delta_b$	$\Delta_a$	$t$	BKS	$\Delta_b$	$\Delta_a$	$t$
R109	<u>833.80</u>	0.00	0.07	81.33	<u>816.21</u>	0.68	0.68	60.13
R110	<u>734.48</u>	0.00	0.00	60.36	<u>729.42</u>	0.17	0.22	74.00
R111	<u>786.59</u>	0.00	0.06	75.77	<u>783.18</u>	0.00	0.03	65.35
R112	<u>658.69</u>	0.00	0.18	142.47	<u>653.03</u>	0.04	0.17	103.17
R201	<u>674.75</u>	0.00	0.00	52.17	<u>674.75</u>	0.00	0.00	52.62
R202	<u>647.17</u>	0.00	0.00	43.95	<u>647.17</u>	0.00	0.00	44.75
R203	<u>571.02</u>	0.00	0.00	98.35	<u>571.02</u>	0.00	0.00	96.64
R204	489.14	0.00	0.00	136.35	489.14	0.00	0.00	150.63
R205	<u>606.43</u>	0.00	0.32	76.85	<u>606.43</u>	0.00	0.32	79.41
R206	<u>579.37</u>	0.00	0.00	84.39	<u>579.37</u>	0.00	0.00	88.82
R207	<u>512.56</u>	0.00	0.00	86.27	<u>512.56</u>	0.00	0.00	85.25
R208	<u>467.16</u>	0.00	0.04	132.02	<u>467.16</u>	0.00	0.04	126.45
R209	<u>602.50</u>	0.00	0.28	177.10	<u>602.50</u>	0.00	0.21	130.89
R210	<u>521.18</u>	0.00	0.00	115.37	<u>519.78</u>	0.00	0.00	113.71
R211	<u>516.53</u>	0.00	0.00	53.26	<u>516.53</u>	0.00	0.00	47.74
RC101	<u>1094.76</u>	0.00	0.02	56.77	<u>1066.48</u>	0.00	0.00	67.50
RC102	<u>913.05</u>	0.00	0.00	46.87	<u>900.71</u>	0.00	0.07	72.36
RC103	<u>857.93</u>	0.00	0.00	46.44	<u>847.00</u>	0.00	0.00	66.04
RC104	<u>694.20</u>	0.00	0.00	53.42	<u>692.05</u>	0.00	0.00	58.29
RC105	<u>1009.44</u>	0.00	0.00	40.65	<u>997.59</u>	0.00	0.00	38.48
RC106	<u>929.72</u>	0.02	0.67	90.35	<u>929.66</u>	0.00	0.15	87.70
RC107	<u>791.52</u>	0.02	0.88	80.58	<u>788.67</u>	0.19	0.22	147.00
RC108	<u>742.30</u>	0.00	0.01	88.47	<u>726.09</u>	0.00	1.32	57.98
RC201	<u>812.23</u>	0.00	0.00	38.94	<u>812.23</u>	0.00	0.00	36.19
RC202	<u>799.71</u>	0.00	0.00	53.19	<u>799.71</u>	0.00	0.00	47.50
RC203	<u>670.03</u>	0.00	0.00	78.86	<u>670.03</u>	0.00	0.00	76.82
RC204	<u>610.24</u>	0.00	0.00	83.16	<u>610.24</u>	0.00	0.00	85.38
RC205	<u>720.66</u>	0.00	0.00	65.03	<u>720.66</u>	0.00	0.00	62.38
RC206	<u>795.20</u>	0.00	0.00	69.50	<u>795.20</u>	0.00	0.00	74.77
RC207	<u>646.78</u>	0.00	0.00	81.45	<u>646.78</u>	0.00	0.00	86.60
RC208	<u>577.33</u>	0.00	0.00	129.35	<u>577.33</u>	0.00	0.00	140.68

TABLE A3 Results of LNS  $\times$  GTS on the 100-customer EVRPTW-S instances

Instance	Full recharges (F)				Partial recharges (P)			
	BKS	$\Delta_b$	$\Delta_a$	$t$	BKS	$\Delta_b$	$\Delta_a$	$t$
C101	<u>1059.65</u>	0.00	0.00	88.43	<u>1025.41</u>	3.09	3.09	91.20
C102	<u>1028.39</u>	0.00	0.00	137.91	<u>1025.41</u>	0.00	0.00	149.62
C103	<u>1014.77</u>	0.16	0.18	316.54	<u>995.85</u>	0.00	0.02	401.69
C104	972.99	0.10	0.15	541.79	954.89	0.00	0.31	451.15
C105	<u>1039.74</u>	0.00	0.00	105.52	<u>1031.24</u>	0.00	0.00	114.17
C106	<u>1033.08</u>	0.00	0.00	124.83	<u>1027.44</u>	0.00	0.00	137.15
C107	<u>1031.45</u>	0.00	0.00	139.21	<u>1030.40</u>	0.00	0.00	116.65
C108	<u>1024.69</u>	0.26	0.51	207.14	<u>1014.81</u>	0.26	0.32	252.07
C109	1006.02	0.00	0.06	350.15	995.79	0.00	0.16	348.80
C201	<u>651.96</u>	0.00	0.00	176.78	<u>630.35</u>	0.00	0.00	180.22
C202	651.58	0.00	0.00	250.44	<u>630.35</u>	0.00	0.00	243.77
C203	646.08	0.00	0.00	292.42	<u>630.35</u>	0.00	0.00	290.65
C204	637.54	0.00	0.00	366.40	629.33	0.00	0.00	361.93
C205	<u>641.53</u>	0.00	0.00	194.39	<u>630.35</u>	0.00	0.00	192.72
C206	<u>638.56</u>	0.00	0.00	209.59	<u>630.35</u>	0.00	0.00	211.58

TABLE A3 Continued

Instance	Full recharges (F)				Partial recharges (P)			
	BKS	$\Delta_b$	$\Delta_a$	$t$	BKS	$\Delta_b$	$\Delta_a$	$t$
C207	<u>638.56</u>	0.00	0.00	227.71	<u>630.35</u>	0.00	0.00	227.94
C208	<u>638.56</u>	0.00	0.00	219.98	<u>630.35</u>	0.00	0.00	218.87
R101	<u>1643.20</u>	0.00	0.19	100.08	<u>1603.06</u>	0.00	0.15	165.49
R102	<u>1475.68</u>	0.00	0.14	370.52	<u>1451.56</u>	0.06	0.28	348.73
R103	<u>1242.25</u>	1.56	1.81	355.84	<u>1210.90</u>	1.08	1.27	331.59
R104	<u>1092.61</u>	0.06	0.22	379.24	<u>1072.88</u>	0.63	0.77	470.07
R105	<u>1395.72</u>	0.50	0.57	228.84	<u>1372.94</u>	0.00	0.12	259.50
R106	<u>1261.96</u>	0.30	0.63	278.50	<u>1251.89</u>	0.03	0.36	368.21
R107	<u>1167.68</u>	0.48	0.75	328.72	<u>1159.73</u>	0.29	0.34	340.70
R108	<u>1025.20</u>	0.11	0.17	544.03	<u>1017.04</u>	0.12	0.15	343.97
R109	<u>1274.85</u>	0.00	0.63	346.28	<u>1257.18</u>	0.50	0.69	308.23
R110	<u>1114.94</u>	0.00	0.64	356.26	<u>1099.15</u>	0.46	0.75	415.22
R111	<u>1133.09</u>	0.05	1.06	407.75	<u>1111.65</u>	0.23	1.10	413.64
R112	<u>1047.80</u>	0.00	1.22	514.74	<u>1038.82</u>	0.07	1.70	497.00
R201	<u>1098.61</u>	0.16	0.36	497.81	<u>1098.24</u>	0.28	0.39	588.01
R202	<u>993.15</u>	0.01	0.20	439.86	<u>993.13</u>	0.14	0.22	505.17
R203	<u>864.84</u>	0.00	0.03	706.86	<u>864.84</u>	0.00	0.06	643.77
R204	<u>720.64</u>	0.00	0.21	790.90	<u>720.64</u>	0.00	0.00	782.05
R205	<u>950.83</u>	0.02	0.05	628.54	<u>950.83</u>	0.10	0.10	478.17
R206	<u>887.91</u>	0.00	0.29	897.54	<u>887.70</u>	0.00	0.34	938.44
R207	<u>792.12</u>	0.00	0.12	510.73	<u>792.12</u>	0.00	0.02	680.69
R208	<u>707.40</u>	0.00	0.00	956.23	<u>707.40</u>	0.00	0.00	791.69
R209	<u>855.58</u>	0.07	0.29	669.66	<u>855.42</u>	0.05	0.27	918.48
R210	<u>830.45</u>	0.00	0.17	811.32	<u>830.45</u>	0.00	0.17	768.04
R211	<u>760.31</u>	0.00	0.20	528.20	<u>759.31</u>	0.00	0.45	655.52
RC101	<u>1738.72</u>	0.01	0.06	245.90	<u>1713.86</u>	0.47	0.77	292.49
RC102	<u>1601.24</u>	0.19	0.61	324.38	<u>1548.77</u>	0.06	0.49	283.45
RC103	<u>1347.77</u>	0.71	1.01	395.68	<u>1314.33</u>	0.74	1.24	396.04
RC104	<u>1219.98</u>	0.22	0.48	444.85	<u>1214.25</u>	0.41	0.97	468.56
RC105	<u>1545.44</u>	0.00	0.14	383.86	<u>1531.76</u>	0.00	0.02	272.43
RC106	<u>1408.89</u>	0.55	0.77	274.63	<u>1396.48</u>	0.06	0.36	376.77
RC107	<u>1258.41</u>	0.60	1.90	384.89	<u>1243.51</u>	0.22	0.38	518.14
RC108	<u>1253.31</u>	0.00	0.45	409.20	<u>1250.32</u>	0.14	0.48	408.39
RC201	<u>1258.27</u>	0.16	0.26	361.57	<u>1258.27</u>	0.00	0.17	395.15
RC202	<u>1139.29</u>	0.00	0.13	268.16	<u>1139.29</u>	0.00	0.09	323.25
RC203	<u>957.22</u>	0.00	0.01	603.23	<u>957.22</u>	0.00	0.01	554.83
RC204	<u>830.11</u>	0.00	0.02	563.82	<u>830.11</u>	0.00	0.00	591.21
RC205	<u>1072.08</u>	0.00	0.00	569.53	<u>1072.08</u>	0.00	0.03	455.31
RC206	<u>1069.39</u>	0.27	0.43	472.50	<u>1069.39</u>	0.00	0.14	671.50
RC207	<u>925.72</u>	0.00	0.00	485.45	<u>925.72</u>	0.00	0.00	430.32
RC208	<u>800.18</u>	0.00	0.00	735.87	<u>800.18</u>	0.00	0.00	677.66

TABLE A4 Performance of LNS  $\times$  GTS on 100-customer instances compared to the previous BKS from Desaulniers et al. [5,6], which are labeled as BKS<sup>†</sup>

Instance	Full recharges (F)			Partial recharges (P)		
	BKS <sup>†</sup>	$\Delta_b$	$\Delta_a$	BKS <sup>†</sup>	$\Delta_b$	$\Delta_a$
C101	<u>1059.65</u>	0.00	0.00	<u>1025.41</u>	3.09	3.09
C102	<u>1028.39</u>	0.00	0.00	<u>1025.41</u>	0.00	0.00
C103	<u>1014.77</u>	0.16	0.18	<u>995.85</u>	0.00	0.02
C104	985.47	-1.16	-1.12	969.57	-1.51	-1.21
C105	<u>1039.74</u>	0.00	0.00	<u>1031.24</u>	0.00	0.00
C106	<u>1033.08</u>	0.00	0.00	<u>1027.44</u>	0.00	0.00

TABLE A4 Continued

Instance	Full recharges (F)			Partial recharges (P)		
	BKS <sup>†</sup>	$\Delta_b$	$\Delta_a$	BKS <sup>†</sup>	$\Delta_b$	$\Delta_a$
C107	<u>1031.45</u>	0.00	0.00	<u>1030.40</u>	0.00	0.00
C108	<u>1024.69</u>	0.26	0.51	<u>1014.81</u>	0.26	0.32
C109	1010.46	-0.44	-0.38	998.96	-0.32	-0.16
C201	<u>651.96</u>	0.00	0.00	<u>630.35</u>	0.00	0.00
C202	660.91	-1.41	-1.41	<u>630.35</u>	0.00	0.00
C203	674.63	-4.23	-4.23	<u>630.35</u>	0.00	0.00
C204	-	-	-	-	-	-
C205	<u>641.53</u>	0.00	0.00	<u>630.35</u>	0.00	0.00
C206	<u>638.56</u>	0.00	0.00	<u>630.35</u>	0.00	0.00
C207	<u>638.56</u>	0.00	0.00	<u>630.35</u>	0.00	0.00
C208	<u>638.56</u>	0.00	0.00	<u>630.35</u>	0.00	0.00
R101	<u>1643.20</u>	0.00	0.19	<u>1603.06</u>	0.00	0.15
R102	<u>1475.68</u>	0.00	0.14	<u>1451.56</u>	0.06	0.28
R103	<u>1242.25</u>	1.56	1.81	<u>1210.90</u>	1.08	1.27
R104	1095.49	-0.21	-0.05	1080.91	-0.12	0.02
R105	<u>1395.72</u>	0.50	0.57	<u>1372.94</u>	0.00	0.12
R106	<u>1261.96</u>	0.30	0.63	<u>1251.89</u>	0.03	0.36
R107	<u>1167.68</u>	0.48	0.75	<u>1159.73</u>	0.29	0.34
R108	<u>1025.20</u>	0.11	0.17	1018.76	-0.05	-0.02
R109	<u>1274.85</u>	0.00	0.63	1257.18	0.50	0.69
R110	1120.48	-0.49	0.14	1118.54	-1.28	-0.99
R111	<u>1133.09</u>	0.05	1.06	<u>1111.65</u>	0.23	1.10
R112	1057.63	-0.93	0.28	1055.69	-1.53	0.07
R201	<u>1098.61</u>	0.16	0.36	<u>1098.24</u>	0.28	0.39
R202	993.15	0.01	0.20	993.13	0.14	0.22
R203	896.11	-3.49	-3.46	890.23	-2.85	-2.79
R204	868.79	-17.05	-16.87	858.12	-16.02	-16.02
R205	<u>950.83</u>	0.02	0.05	<u>950.83</u>	0.10	0.10
R206	946.05	-6.15	-5.87	<u>887.70</u>	0.00	0.34
R207	878.22	-9.80	-9.69	876.73	-9.65	-9.63
R208	-	-	-	-	-	-
R209	862.85	-0.78	-0.56	862.85	-0.81	-0.60
R210	894.96	-7.21	-7.05	894.96	-7.21	-7.05
R211	892.42	-14.80	-14.63	892.42	-14.92	-14.53
RC101	1738.72	0.01	0.06	<u>1713.86</u>	0.47	0.77
RC102	1601.24	0.19	0.61	<u>1548.77</u>	0.06	0.49
RC103	<u>1347.77</u>	0.71	1.01	<u>1314.33</u>	0.74	1.24
RC104	<u>1219.98</u>	0.22	0.48	1228.67	-0.77	-0.21
RC105	1546.65	-0.08	0.07	1538.68	-0.45	-0.43
RC106	<u>1408.89</u>	0.55	0.77	1401.10	-0.27	0.03
RC107	<u>1258.41</u>	0.60	1.90	1258.41	-0.97	-0.81
RC108	<u>1253.31</u>	0.00	0.45	1253.31	-0.10	0.24
RC201	<u>1258.27</u>	0.16	0.26	<u>1258.27</u>	0.00	0.17
RC202	<u>1139.29</u>	0.00	0.13	<u>1139.29</u>	0.00	0.09
RC203	1032.46	-7.29	-7.27	<u>957.22</u>	0.00	0.01
RC204	-	-	-	-	-	-
RC205	<u>1072.08</u>	0.00	0.00	<u>1072.08</u>	0.00	0.03
RC206	<u>1069.39</u>	0.27	0.43	<u>1069.39</u>	0.00	0.14
RC207	<u>925.72</u>	0.00	0.00	<u>925.72</u>	0.00	0.00
RC208	-	-	-	-	-	-
All		-1.33	-1.13		-0.99	-0.81